|  |  |
| --- | --- |
| **Sl.**  **No.** | **Name of Experiment** |
| 1. | 1.Installation of Python, Django and Visual Studio code editors can be demonstrated.  2.Creation of virtual environment, Django project and App should be demonstrated  3.Develop a Django app that displays current date and time in server   1. 4.Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server. |
| 2. | 1. 5.Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event 2. 6.Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages: contact us, About Us and Home page of any website.   7.Develop a Django app that performs student registration to a course. It should also  display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field. |
| 3. | 1. 8.For student and course models created in Lab experiment for Module2, register admin interfaces, perform migrations and illustrate data entry through admin forms.   9.Develop a Model form for student that contains his topic chosen for project, languages  used and duration with a model called project. |
| 4. | 1. 10.For students enrolment developed in Module 2, create a generic class view which displays list of students and detailview that displays student details for any selected student in the list.   11. Develop example Django app that performs CSV and PDF generation for any models  created in previous laboratory component. |
| 5. | 1. 12. Develop a registration page for student enrolment as done in Module 2 but without page refresh using AJAX.   13. Develop a search application in Django using AJAX that displays courses enrolled by a student being searched. |

**7 8 9 n 10 pgms**

**Copy to base.html**

<!DOCTYPE html>

<html>

<head>

<title>Student Registration</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>

</head>

<body>

{% block content %}

{% endblock %}

<!-- Bootstrap JavaScript (Optional, if you need Bootstrap JavaScript features) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV56w/C4QV8q+iJ6W/5pScbXB+SFZ5qONbWJcZ1QxIhmmj5fDDZ9OvhSjJ2Ks2" crossorigin="anonymous"></script>

</body>

</html>

**Copy to register\_student.html**

{% load static %}

<!DOCTYPE html>

<html>

<head>

<title>Student Registration</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<script src="{% static 'js/jquery.min.js' %}"></script>

</head>

<body>

<div class="container">

<h1>Student Registration</h1>

<form id="student-form" method="post">

{% csrf\_token %}

{% for field in form %}

<div class="form-group">

{{ field.label\_tag }}

{{ field }}

{% if field.help\_text %}

<small class="form-text text-muted">{{ field.help\_text }}</small>

{% endif %}

{% for error in field.errors %}

<div class="alert alert-danger">{{ error }}</div>

{% endfor %}

</div>

{% endfor %}

<button type="submit" class="btn btn-primary">Submit</button>

</form>

<div id="response"></div>

</div>

<script>

$(document).ready(function() {

$('#student-form').on('submit', function(e) {

e.preventDefault();

var formData = $(this).serialize();

$.ajax({

type: 'POST',

url: '{% url "student" %}',

data: formData,

success: function(response) {

if (response.success) {

$('#response').html('<div class="alert alert-success">Student registered successfully.</div>');

$('#student-form')[0].reset();

} else {

$('#response').html('<div class="alert alert-danger">An error occurred. Please try again.</div>');

}

},

error: function(xhr, errmsg, err) {

$('#response').html('<div class="alert alert-danger">An error occurred: ' + xhr.status + ' ' + xhr.responseText + '</div>');

}

});

});

});

</script>

</body>

</html>

**Copy to forms.py**

from django import forms

from .models import Student

class StudentForm(forms.ModelForm):

class Meta:

model = Student

fields = ['first\_name', 'last\_name', 'email']

widgets = {

'first\_name': forms.TextInput(attrs={'class': 'form-control'}),

'last\_name': forms.TextInput(attrs={'class': 'form-control'}),

'email': forms.EmailInput(attrs={'class': 'form-control'}),

}

**Copy to models.py**

from django.db import models

# Create your models here.

class Student(models.Model):

first\_name = models.CharField(max\_length=255, default='')

last\_name = models.CharField(max\_length=255, default='')

email = models.EmailField(unique=True, default='')

**Copy to index.html**

{% extends 'base.html' %}

{% block content %}

<div class="container">

<h1 class="mt-5">Courses</h1>

<ul class="list-group mt-3">

{% for course in courses %}

<li class="list-group-item">

<a href="{% url 'student\_list' course.id %}">{{ course.name }}</a>

</li>

{% endfor %}

</ul>

<h1 class="mt-5 my-5">Projects</h1>

<ul class="list-group mt-3">

{% for project in projects %}

<li class="list-group-item">

<a href="{% url 'project\_student\_list' project.id %}">{{ project.topic }}</a>

</li>

{% endfor %}

</ul>

<h1 class="mt-5 my-5">Students</h1>

<ul class="list-group mt-3">

{% for student in students %}

<li class="list-group-item">

<a href="{% url 'student\_detail' student.pk %}">{{ student.first\_name }} {{ student.last\_name }}</a>

</li>

{% endfor %}

</ul>

<a class="btn btn-primary mt-3" href="{% url 'register\_student' %}">Register Student</a>

<a class="btn btn-success mt-3" href="{% url 'register\_course' %}">Register Course</a>

<a class="btn btn-primary mt-3" href="{% url 'register\_project' %}">Register Project</a>

{% endblock %}

**Copy to project\_detail.html**

{% extends 'base.html' %}

{% block content %}

<div class="container">

<h1 class="mt-5">Project Details</h1>

<div class="card mt-3">

<div class="card-body">

<h5 class="card-title">{{ project.name }}</h5>

<p class="card-text">Topic: {{ project.topic }}</p>

<p class="card-text">Languages Used: {{ project.languages\_used }}</p>

<p class="card-text">Duration: {{ project.duration }}</p>

</div>

</div>

</div>

{% endblock %}

**Copy to register\_course.html**

{% extends 'base.html' %}

{% load static %}

{% block content %}

<h1 style="text-align: center;" class="my-5">Register Course</h1>

<div class="container">

<div class="row justify-content-center">

<div class="col-md-6">

<form method="post">

{% csrf\_token %}

{% for field in form %}

<div class="form-group">

{{ field.label\_tag }}

{{ field }}

{% if field.help\_text %}

<small class="form-text text-muted">{{ field.help\_text }}</small>

{% endif %}

{% for error in field.errors %}

<div class="alert alert-danger">{{ error }}</div>

{% endfor %}

</div>

{% endfor %}

<button type="submit" class="btn btn-primary text-center text-center">Register</button>

</form>

</div>

</div>

</div>

{% endblock %}

**Copy to register\_project.html**

<!-- create\_project.html -->

{% extends 'base.html' %}

{% block content %}

<div class="container my-3">

<h1>Create Project</h1>

<form method="post">

{% csrf\_token %}

<div class="form-group">

<label for="id\_topic">Topic:</label>

{{ form.topic }}

</div>

<div class="form-group">

<label for="id\_languages\_used">Languages Used:</label>

{{ form.languages\_used }}

</div>

<div class="form-group">

<label for="id\_duration">Duration:</label>

{{ form.duration }}

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

{% endblock %}

**Copy to register\_student.html**

{% extends 'base.html' %}

{% block content %}

<div class="container my-3">

<h2>Student Registration</h2>

<form method="post" class="needs-validation" novalidate>

{% csrf\_token %}

{{ form.non\_field\_errors }}

<div class="form-group">

{{ form.first\_name.label\_tag }}

{{ form.first\_name }}

{{ form.first\_name.errors }}

</div>

<div class="form-group">

{{ form.last\_name.label\_tag }}

{{ form.last\_name }}

{{ form.last\_name.errors }}

</div>

<div class="form-group">

{{ form.email.label\_tag }}

{{ form.email }}

{{ form.email.errors }}

</div>

<div class="form-group">

{{ form.courses.label\_tag }}

{{ form.courses }}

{{ form.courses.errors }}

</div>

<div class="form-group">

{{ form.project.label\_tag }}

{{ form.project }}

{{ form.project.errors }}

</div>

<button type="submit" class="btn btn-primary">Register</button>

</form>

</div>

{% endblock %}

**Copy to Stu\_list.html**

{% extends 'base.html' %}

{% block content %}

<h1>Students</h1>

<ul>

{% for student in students %}

<li>

<a href="{% url 'student\_detail' student.pk %}">{{ student.first\_name }} {{ student.last\_name }}</a>

</li>

{% endfor %}

</ul>

{% endblock %}

**Copy to Student\_detail.html**

{% extends 'base.html' %}

{% block content %}

<div class="container my-5">

<div class="row">

<div class="col-md-6 offset-md-3">

<div class="card">

<div class="card-body">

<h1 class="card-title"> Name: {{ student.first\_name }} {{ student.last\_name }}</h1>

<h1 class="card-title">Email: {{ student.email }}</h1>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

**Copy to Student\_list.html**

{% extends 'base.html' %}

{% block content %}

<h1>Students Registered for {{ course.name }}</h1>

<ul>

{% for student in students %}

<li>{{ student.first\_name }} {{ student.last\_name }}</li>

{% endfor %}

</ul>

{% endblock %}

**Copy to Student\_list1.html**

{% extends 'base.html' %}

{% block content %}

<h1>Students Registered for {{ project.topic }}</h1>

<ul>

{% for student in students %}

<li>{{ student.first\_name }} {{ student.last\_name }}</li>

{% endfor %}

</ul>

{% endblock %}

**Copy to forms.py**

from django import forms

from .models import Student, Course, Project

class StudentForm(forms.ModelForm):

class Meta:

model = Student

fields = ['first\_name', 'last\_name', 'email', 'courses', 'project'] # Use 'project' instead of 'projects'

widgets = {

'first\_name': forms.TextInput(attrs={'class': 'form-control'}),

'last\_name': forms.TextInput(attrs={'class': 'form-control'}),

'email': forms.EmailInput(attrs={'class': 'form-control'}),

'courses': forms.SelectMultiple(attrs={'class': 'form-select', 'data-toggle': 'dropdown'}),

'project': forms.SelectMultiple(attrs={'class': 'form-select', 'data-toggle': 'dropdown'}), # Use 'project' instead of 'projects'

}

class CourseForm(forms.ModelForm):

class Meta:

model = Course

fields = ['name', 'description']

widgets = {

'name': forms.TextInput(attrs={'class': 'form-control'}),

'description': forms.Textarea(attrs={'class': 'form-control', 'rows': 3}),

}

class ProjectForm(forms.ModelForm):

class Meta:

model = Project

fields = ('topic', 'languages\_used', 'duration')

widgets = {

'topic': forms.TextInput(attrs={'class': 'form-control'}),

'languages\_used': forms.TextInput(attrs={'class': 'form-control'}),

'duration': forms.TextInput(attrs={'class': 'form-control'}),

}

**Copy to models.py**

from django.db import models

class Course(models.Model):

name = models.CharField(max\_length=255)

description = models.TextField(blank=True, null=True)

def \_\_str\_\_(self):

return self.name

class Student(models.Model):

first\_name = models.CharField(max\_length=255, default='')

last\_name = models.CharField(max\_length=255, default='')

email = models.EmailField(unique=True, default='')

courses = models.ManyToManyField(Course, related\_name='students', blank=True)

project = models.ManyToManyField('Project', related\_name='students', blank=True)

def \_\_str\_\_(self):

return f"{self.first\_name} {self.last\_name}"

class Project(models.Model):

topic = models.CharField(max\_length=100, default='') # Added default value for topic

languages\_used = models.CharField(max\_length=100, default='')

duration = models.IntegerField(default=0)

def \_\_str\_\_(self):

return self.topic

<body>

<h1 class="text-center my-4">Export Books</h1>

<div class="container">

<div class="row">

<div class="col-md-6 mb-4">

<div class="card h-100 shadow">

<div class="card-body d-flex flex-column justify-content-center align-items-center">

<i class="fas fa-file-pdf card-icon text-danger mb-3"></i>

<h5 class="card-title">Export as PDF</h5>

<a href="{% url 'export\_books\_pdf' %}" class="btn btn-danger mt-3">Export</a>

</div>

</div>

</div>

<div class="col-md-6 mb-4">

<div class="card h-100 shadow">

<div class="card-body d-flex flex-column justify-content-center align-items-center">

<i class="fas fa-file-csv card-icon text-success mb-3"></i>

<h5 class="card-title">Export as CSV</h5>

<a href="{% url 'export\_books\_csv' %}" class="btn btn-success mt-3">Export</a>

</div>

</div>

</div>

</div>

</div>

<script src="https://kit.fontawesome.com/your-font-awesome-kit.js" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

**Copy to Views**

from django.shortcuts import render, redirect, get\_object\_or\_404

from .models import Course, Student,Project

from .forms import StudentForm, CourseForm,ProjectForm

from django.views.generic import ListView, DetailView

from .models import Course, Student, Project

from django.http import JsonResponse

from django.views.decorators.csrf import csrf\_exempt

def index(request):

courses = Course.objects.all()

projects = Project.objects.all()

students = Student.objects.all()

return render(request, 'registration/index.html', {

'courses': courses,

'projects': projects,

'students': students,

})

def register\_student(request):

if request.method == 'POST':

form = StudentForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

else:

form = StudentForm()

return render(request, 'registration/register\_student.html', {'form': form})

def register\_course(request):

if request.method == 'POST':

form = CourseForm(request.POST)

if form.is\_valid():

form.save()

return redirect('index')

else:

form = CourseForm()

return render(request, 'registration/register\_course.html', {'form': form})

def student\_list(request, course\_id):

course = Course.objects.get(id=course\_id)

students = course.students.all()

return render(request, 'registration/student\_list.html', {'students': students, 'course': course})

def register\_project(request):

if request.method == 'POST':

form = ProjectForm(request.POST)

if form.is\_valid():

project = form.save()

# Redirect to a success page or another view

return redirect('index')

#return redirect('project\_detail', project\_id=project.pk)

else:

form = ProjectForm()

return render(request, 'registration/register\_project.html', {'form': form})

def project\_detail(request, project\_id):

project = get\_object\_or\_404(Project, pk=project\_id)

return render(request, 'registration/project\_details.html', {'project': project})

def project\_student\_list(request, project\_id):

project = Project.objects.get(id=project\_id)

students = project.students.all()

return render(request, 'registration/student\_list.html', {'students': students, 'project': project})

def student\_list\_new(request):

students = Student.objects.all()

return render(request, 'registration/stu\_list.html', {'students': students})

class StudentListView(ListView):

model = Student

template\_name = 'registration/stu\_list.html'

context\_object\_name = 'students'

class StudentDetailView(DetailView):

model = Student

template\_name = 'registration/student\_detail.html'

context\_object\_name = 'student'

from django.shortcuts import render

from django.http import JsonResponse

from .models import Student

def search\_students(request):

is\_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

if is\_ajax:

query = request.GET.get('query', '')

students = Student.objects.filter(first\_name\_\_icontains=query) | Student.objects.filter(last\_name\_\_icontains=query)

results = []

for student in students:

student\_data = {

'id': student.id,

'name': f"{student.first\_name} {student.last\_name}",

'email': student.email,

'courses': [course.name for course in student.courses.all()]

}

results.append(student\_data)

return JsonResponse({'results': results})

return render(request, 'registration/search.html')

**Copy to urls.py under application file**

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

path('register-student/', views.register\_student, name='register\_student'),

path('register-course/', views.register\_course, name='register\_course'),

path('student-list/<int:course\_id>/', views.student\_list, name='student\_list'),

path('projects\_register/', views.register\_project, name='register\_project'),

path('project\_detail/<int:project\_id>/', views.project\_detail, name='project\_detail'),

path('project\_student\_list/<int:project\_id>/', views.project\_student\_list, name='project\_student\_list'),

path('students/', views.StudentListView.as\_view(), name='student\_list\_all'),

path('students/<int:pk>/', views.StudentDetailView.as\_view(), name='student\_detail'),

path('search/', views.search\_students, name='search\_students'),

#path('student-list1/<int:project\_id>/', views.student\_list1, name='student\_list1'),

]

**Under project file**

**1] settings.py**

Under installed app

Last line –‘mention app name’,

**2] urls.py**

Mention this alteration

Path, include

Path(‘’, include(‘app name.urls’)),

Commands used to create database n run program here

**Python3 manage.py makemigrations**

**Python3 manage.py migrate**

**Python3 manage.py runserver**